



Agile Planning and Estimation–Chapter 3

Structure Of Presentation

- Introduction
- User Stories and Epics
- INVEST Model
- Story Points and Planning Poker
- Release Planning and Roadmaps
- Velocity and Capacity Planning

What Are User Stories?

- A **User Story** is a **short, simple description of a feature or requirement** told from the perspective of the end user (customer or stakeholder).
- It is a **core element** in Agile used to capture **what the user wants** and **why** they want it.

Structure of User Story

- A good user story contains 4 major segments to describe the requirements and their testing scenarios.

- 1. Role**
- 2. Goal**
- 3. Benefits**
- 4. Description**
- 5. Acceptance Criteria**

Key Elements of a User Story:

Element	Description
Title	Short name of the feature or goal
Role	Who wants it? (user, admin, customer)
Goal	What do they want to do?
Benefit	Why do they want it? What's the value?

Term	Meaning
Epic	Large user story, broken into smaller stories
Feature	Collection of related user stories
Acceptance Criteria	Conditions that must be true for the story to be "done"

- **Standard Format:**

As a [Role],
I want [Goal],
so that [Benefits/Expected output].

- **Example:**

As a student,
I want **to view my attendance report**,
so that **I can track my class participation**.

Sample Story Cards

As a <Role>
I Want <Goal>
So That <Benefit>

Description....

Acceptance Criteria

As a <Who>
I Want <What>
So That <Why>

Description....

Acceptance Criteria

As a <User Type>
I Want <Function>
So That <Benefit>

Description....

Acceptance Criteria

As a <Situation>
I Want <Motivation>
So That <Expected OutCome>

Description....

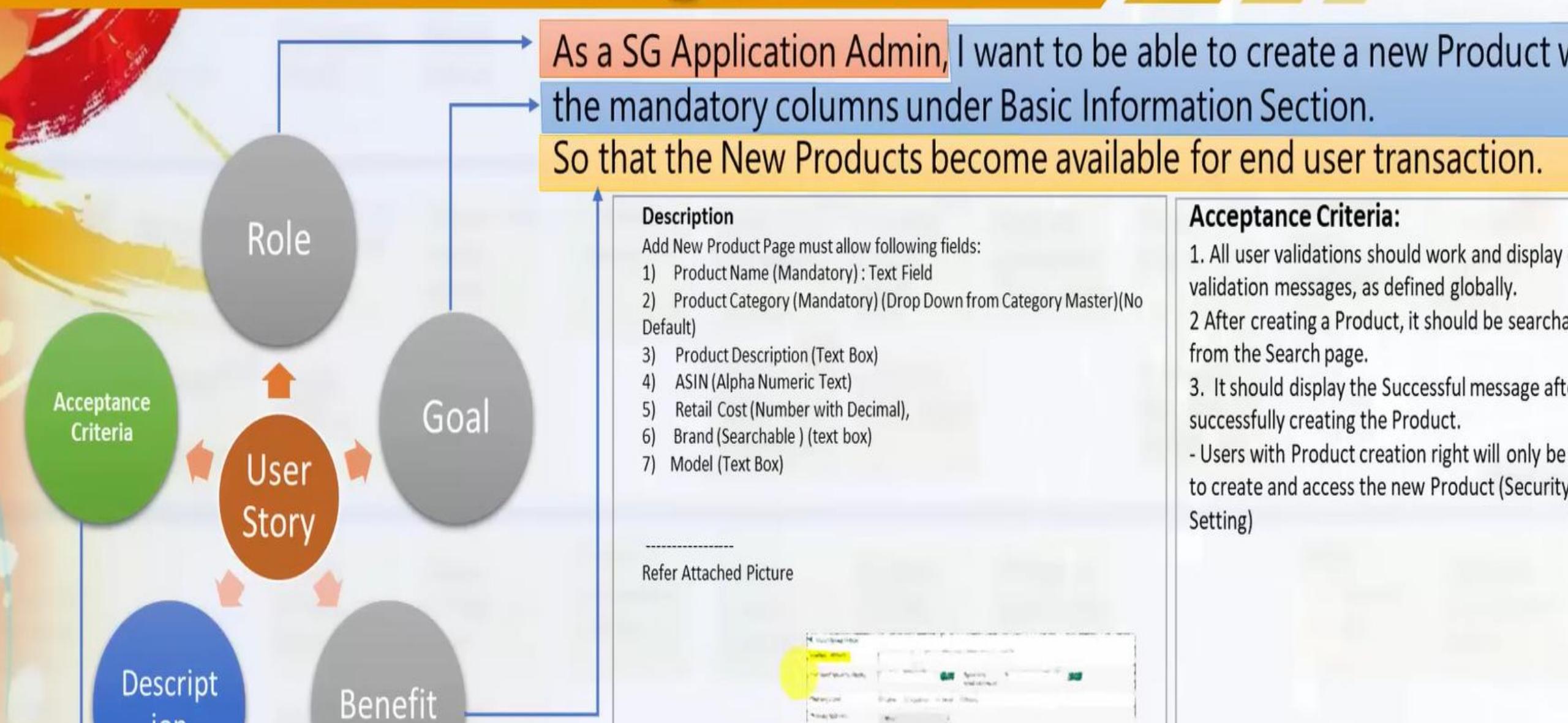
Acceptance Criteria

Agile Planning and Estimation



A	B	C	D	E	F	G	H	I	J	
1	ID	User Story Title	As a (Role)	I want to (Goal)	So that (Benefit)	Acceptance Criteria	Priority	Status	Sprint	Estimate (Story Points)
2	US01	View Attendance Report	Student	View my attendance report	Track my class participation	1. Displays all subjects 2. Shows % attendance per subject	High	To Do	Sprint 1	3
3	US02	Mark Attendance	Teacher	Mark student attendance	Maintain daily attendance records	1. Select date 2. Select present/absent for each student	High	In Progress	Sprint 1	5
4	US03	Login to Portal	User	Login securely	Access personalized dashboard	1. Valid credentials required 2. Forgot password option	Medium	Done	Sprint 1	2
5	US04	Download Report Card	Parent	Download report card	Track child's academic progress	1. PDF download 2. Includes grades and comments	Low	To Do	Sprint 2	3
6										

Understanding 3C

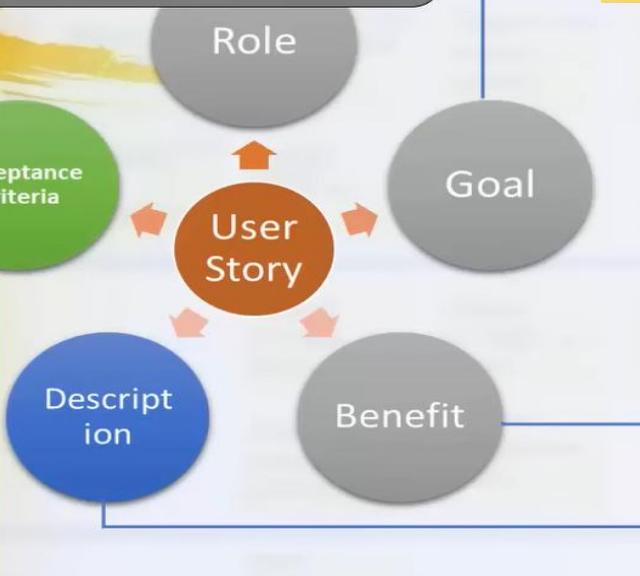


What is 3C in Agile?

Term	Meaning
Card	A short statement (usually a user story) written on a card or in a tracking tool.
Conversation/Composition	The discussion between stakeholders (e.g., developers and users) to clarify the card.
Confirmation	Acceptance criteria that confirm the feature is done correctly.

Understanding 3C

CARD



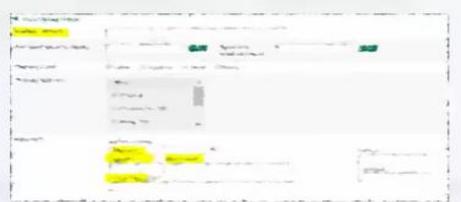
As a SG Application Admin, I want to be able to create a new Product with the mandatory columns under Basic Information Section. So that the New Products become available for end user transaction.

Description:
Add New Product Page must allow following fields:
1) Product Name (Mandatory) : Text Field
2) Product Category (Mandatory) (Drop Down from Category Master)(No Default)
3) Product Description (Text Box)
4) ASIN (Alpha Numeric Text)
5) Retail Cost (Number with Decimal),
6) Brand (Searchable) (text box)
7) Model (Text Box)

Acceptance Criteria:

1. All user validations should work and display correct validation messages, as defined globally.
- 2 After creating a Product, it should be searchable from the Search page.
3. It should display the Successful message after successfully creating the Product.
- Users with Product creation right will only be allowed to create and access the new Product (Security Setting)

Refer Attached Picture



COMPOSITION

CONFIRMATION

Agile Digest

INVEST Criteria for Good User Stories:

Letter	Meaning	Explanation
I	Independent	Can be developed and delivered separately
N	Negotiable	Not a detailed contract, open for discussion
V	Valuable	Delivers value to the user
E	Estimable	Can be estimated in terms of time/effort
S	Small	Small enough to be completed in 1 sprint
T	Testable	Can be verified with tests

Characteristics of User Story

I	N	V	E	S	T
Independent	Negotiable	Valuable Verticals	Estimable	Small	Testable

User Story Should be self contained, vary so that it is no ent dependencies on other story.

User Stories are not explicit contracts, and should leave space for discussion

User Stories must deliver values to the stakeholders or business

You must be able to estimate the user story, once it is ready and groomed.

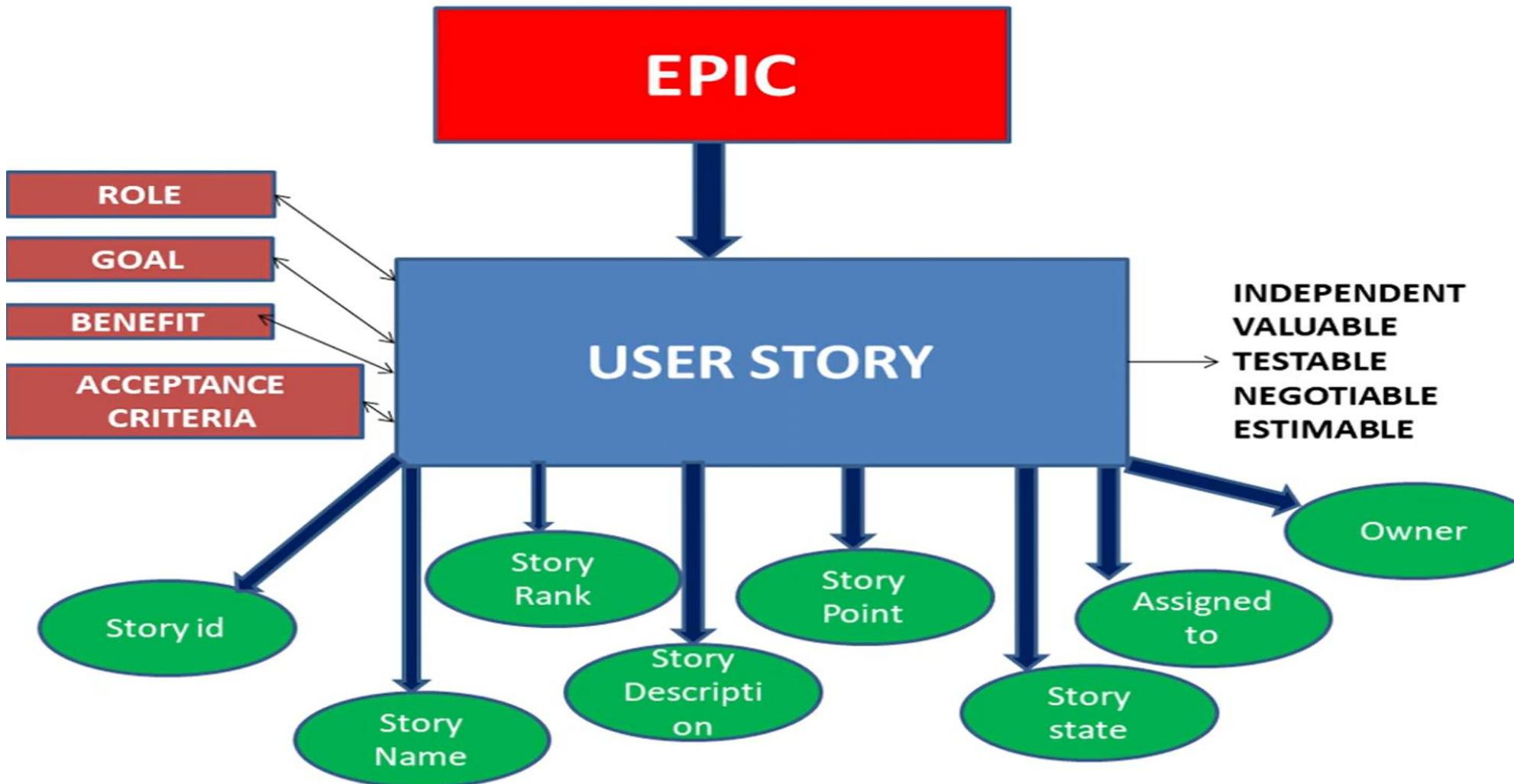
User Story Should not be so big as to become impossible to plan, task out, prioritize with certainty

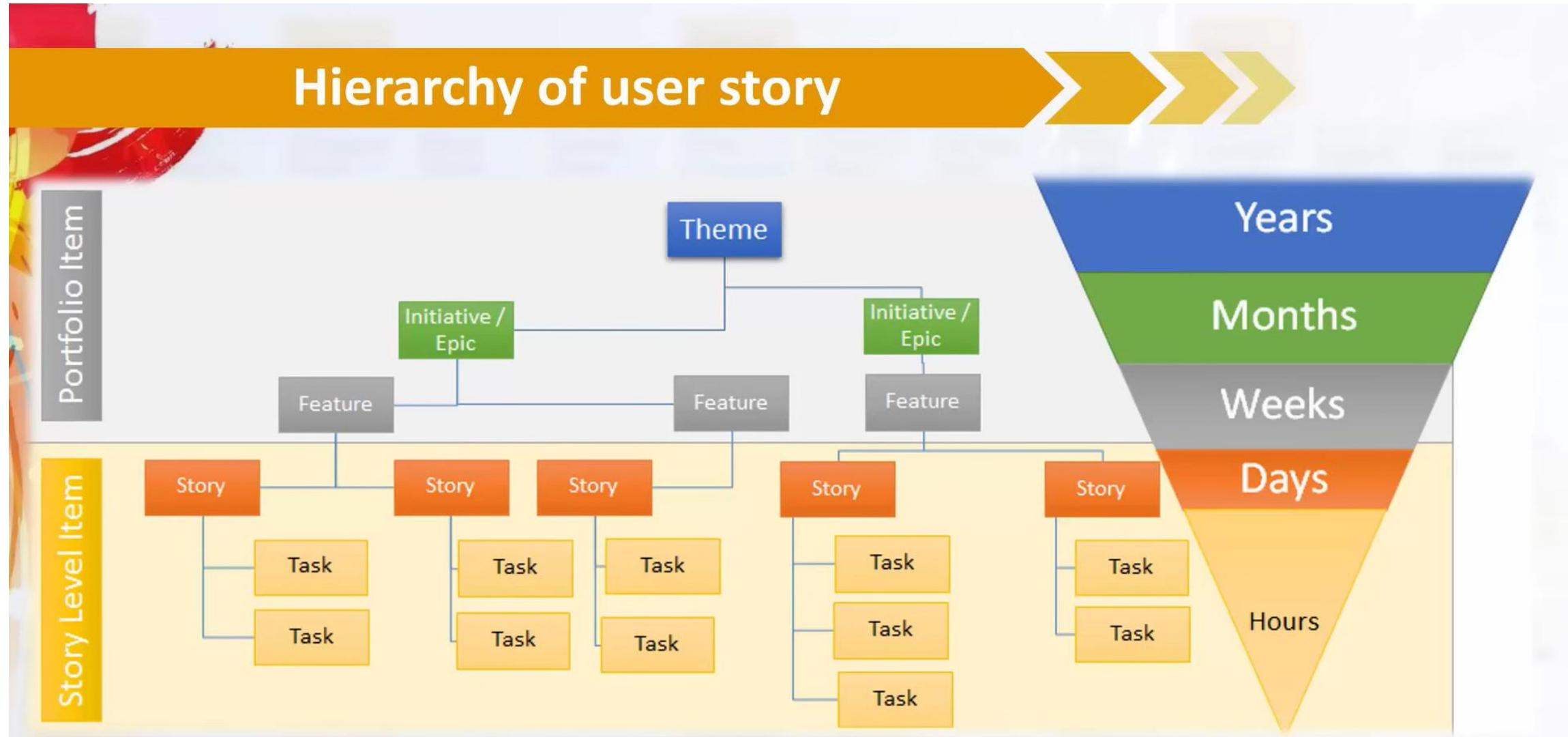
The User Story or its related description must provide the necessary information to make test development possible

- **epic**: Major objective or user need, often cross-cutting several product functions.
- **Feature**: Specific product capability or enhancement, contributing to achieving part of the epic.
- **User Story**: Concise description of a functionality from the user's point of view, contributing to a feature.



- For example, the epic "Mobile App Redesign"
- could be broken down into **features** for
 1. "Design new user interface,"
 2. "Create new navigation paths," and
 3. "Ensure responsiveness on different screen sizes".
- **User stories for feature 1:**
 - i. As user I want to view my contact details so that I can call
 - ii. As user I can update the contact detail with image so that I can see photo while calling





What is Estimation in Agile?

- Agile estimation is the process for estimating the effort required to complete a task in the product backlog.

Types of ESTIMATION

1. Absolute Estimation.
2. Relative Estimation

ABSOLUTE ESTIMATION – TRADITIONAL APPROACH

- Absolute Estimation is done using a fixed unit of measurement.
- Each story is estimated independently in No of hours or no of days only by the members who is working on it.



UNKNOWN FACTORS

- Technical Uncertainties
- Emergency Leave of john
- Learning Time

John (Dev) commits to complete **User Story** in 5 hours



John could not complete the **User story 1** in 5 Hours

User Story :

AS A User
I WANT to have online Student enquiry form
SO THAT I can get enquiries from students

Members	Estimation
John Developer (Dev)	5 Hours
Mary (QA)	1 Hour
Total	6 Hours

- Effects the Sprint Goal
- Effects the Product Delivery
- Not finishing task within time will generate stress to the developer who estimated the story

What are Story Points?

- Story points are a relative measure of effort required to complete a task in Agile.
- Not time-based (e.g., hours or days).
- Helps plan and manage work in Agile teams.
- They are not hours but a complexity score that considers:
 1. Amount of Work
 2. Complexity
 3. Risk & Uncertainty

Why Use Story Points?

- To estimate effort without the bias of time.
- Focuses on complexity and uncertainty.
- Enables better sprint planning.
- Encourages team collaboration.

Factors Influencing Story Points

- **Complexity:** How hard is the task technically?
- **Risk/Uncertainty:** Are there unknowns?
- **Amount of Work:** How much work is needed?
- **Dependencies:** Are other tasks involved?

Common Story Point Scale

- Typically follows the Fibonacci sequence:
- 0, 1, 1, 2, 3, 5, 8, 13, 21, ...
- Reflects increasing uncertainty as size grows.

Example: Estimating Story Points

- User Story A: Login (**2 SP**) - Standard form
- User Story B: Reset Password (**5 SP**) - Email integration
- User Story C: Upload Profile Pic (**8 SP**) - Image cropping
- User Story D: Admin Dashboard (**13 SP**) - Charts, filters

RELATIVE ESTIMATION

Relative estimation is the process of estimating task completion, not by units of time or no of days, but rather by comparing with other tasks in terms of :

- Complexity:
- Risk:
- Implementation:
- Deployment:
- Interdependencies:

Advantages

- More flexible in nature
- No Single person will be responsible for estimation
- It gives chance for requirement changes
- It gives time for Learning due to technical uncertainties

Each Task is estimated by all the members in the team.



1 point



3 points



5 points



8 points



11 points

RELATIVE ESTIMATION - AGILE



User Story 1 :

AS A User(Student)
I WANT to have a Login Page
SO THAT I can use School Management software successfully

User Stories	Estimation
User Story 1	2 Story Points
User Story 2	

RELATIVE ESTIMATION - AGILE



Peter-Dev

Mark-Dev

Mary-Dev

Amy-Dev

Ram-QA

User Story 2 :

AS A User(Student)
I WANT to have a payment functionality
SO THAT I can pay my school fees in School Software

User Stories	Estimation
User Story 1	2 Story Points
User Story 2	8 Story Points

Assumptions:

User Story : Requirement

As a customer,

*I want to add products to my shopping cart
so that I can purchase item during checkout*

Complexity to Calculate

- UI Development*
- Business Logic*
- Database Integration*
- Testing*

Complexity Aggregation

- Addition*

*We can add or average our complexities to get
the story points, In this example we will add
the complexity to align it with a bucket*

Complexity Levels and Values

S = 1, L = 2, M = 3, L = 4

Each Estimator will estimate their own story point and most voted story point will be assigned to the user story

Assumptions:

UI Design

- Small [1]
- Low [2]
- Medium [3]
- High [4]

Business Logic

- Small [1]
- Low [2]
- Medium [3]
- High [4]

DB Integration

- Small [1]
- Low [2]
- Medium [3]
- High [4]

Manual testing

- Small [1]
- Low [2]
- Medium [3]
- High [4]

Total : UI Design + Business Logic + DB Integration + Testing = 1+2+0+1 = 4



Notes:

- You can create your areas of complexity as per your needs. You can add areas like Analysis or ETL Job, or Automation testing.
- You can increase the level of complexity for your needs.
- Each Team member, except the Scrum Master and Product Owner, can use these techniques to gauge the complexity of the story and vote with the story point during estimation. The most voted story point will be marked for the user story.
- You can build an application to do your calculations or simply use a pen and paper.

ETL stands for Extract, Transform, and Load and represents the backbone of data engineering where data gathered from different sources is normalized and consolidated for the purpose of analysis and

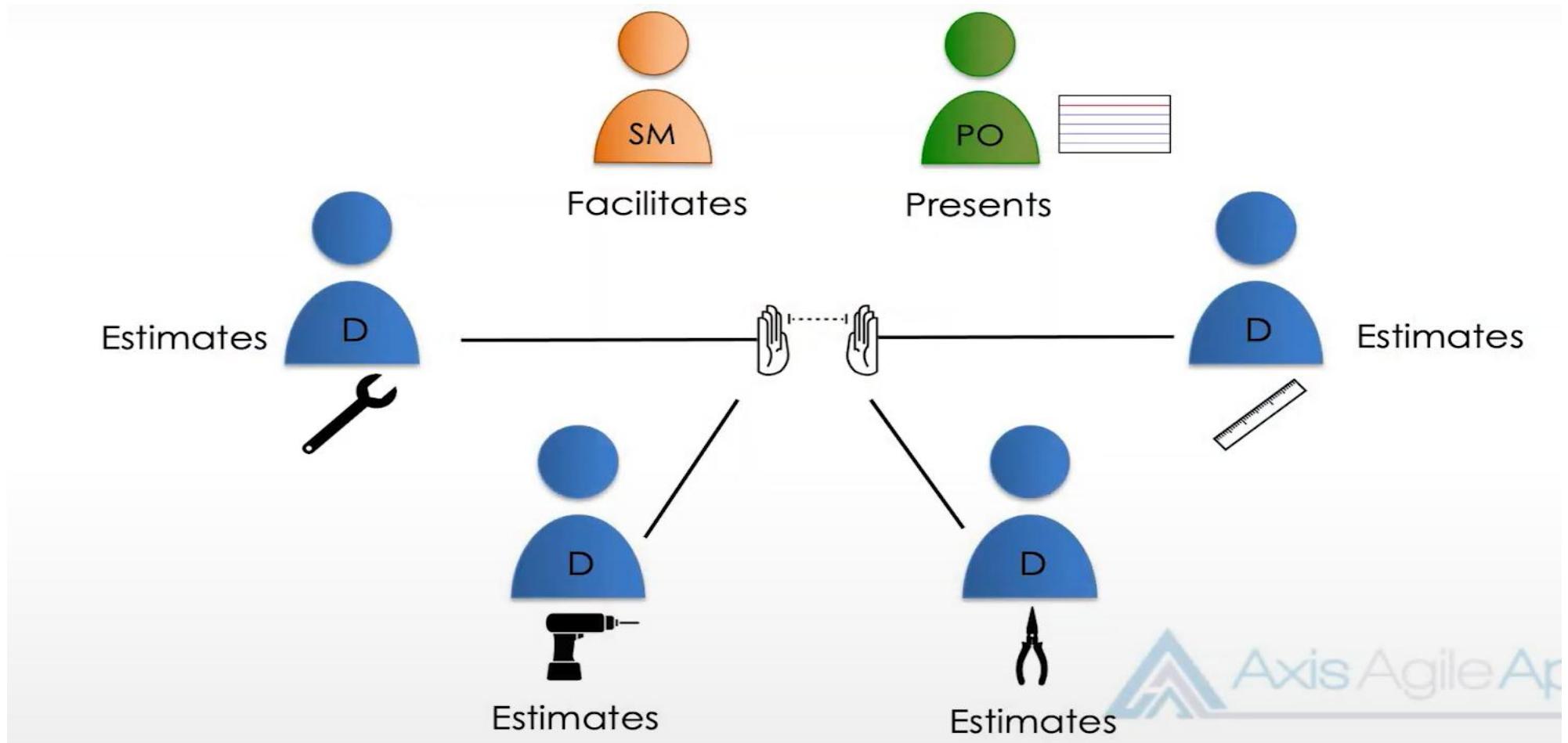
Planning Poker

Potential levels of estimation

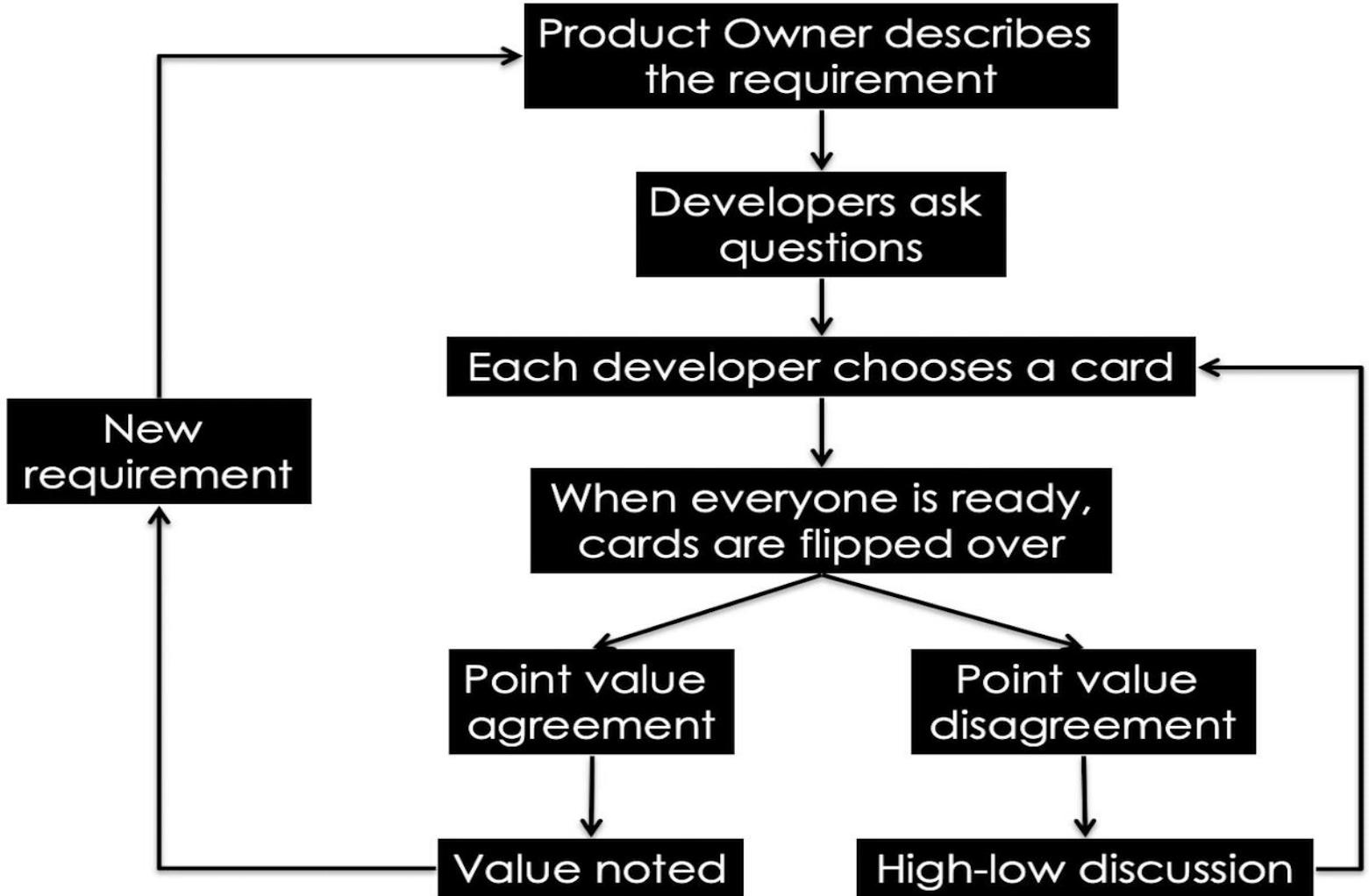


Where	Level of Detail	Units	Used For
Sprint Backlog	Detailed Estimates	Time	Sprint Forecast
Product Backlog	High-Level Estimates	Points ("Story Points")	Release Planning

Who attends?



Planning Poker process:



Planning Poker Process

- **Read the User Story** – Each team member understands the requirement.
- **Discuss Assumptions** – Clarify doubts.
- **Select a Card (Story Point)** – Each member privately chooses an estimation card.
- **Reveal Cards Together** – Discuss differences if estimates vary.
- **Agree on Final Estimate** – Assign the agreed story point value.

Planning Poker:

Example Context – Attendance Database:

We want to design and implement an Attendance Database for a college system.

The team uses Planning Poker to assign story points.

User Story	Complexity	Estimated Story Points
US1: As an admin, I want to add a new student record so that attendance can be tracked.	Low	2 SP
US2: As a faculty member, I want to mark attendance for students in bulk so I can save time.	Medium	5 SP
US3: As a faculty member, I want to edit attendance records so that mistakes can be corrected.	Medium	5 SP

Planning Poker: Example Context – Attendance Database:

User Story	Complexity	Estimated Story Points
US4: As a student, I want to view my attendance percentage so I can track my presence in class.	Medium-High	8 SP
US5: As an admin, I want to generate monthly attendance reports so I can submit them to management.	High	13 SP
US6: As a system, I want to notify students when attendance drops below 75% so they can take action.	High	13 SP

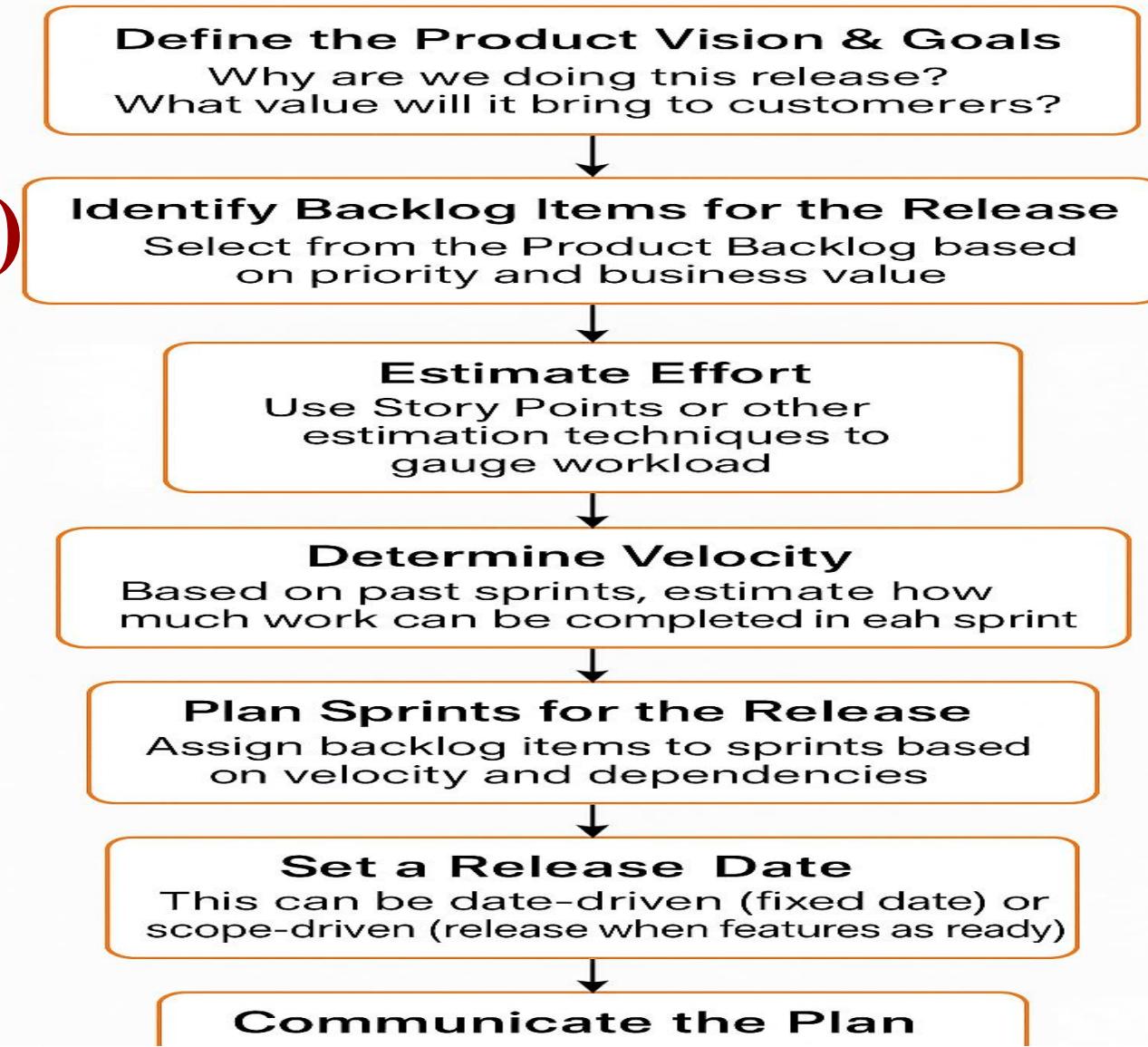
What is Release Planning?

- Release planning is the process of **deciding what features, enhancements, or fixes** will be delivered in a specific product release and when they will be available to customers.
- It connects the high-level product vision with the detailed sprint plans.
- Think of it as:
"What will we deliver, in what order, and by which release date?"

Key Objectives of Release Planning

- Define the **scope** of the release (what's included).
- Set a **timeline** or expected delivery date.
- Allocate **resources** and set priorities.
- Align stakeholders (Product Owner, Scrum Master, Development Team, Business Owners).
- Reduce uncertainty by providing a **roadmap**.

Release Planning Process (Step-by-Step)



Priority	Scope	Size (Story Points)		
Must-Have	Feature 1	8		
Must-Have	Feature 2	5		
Must-Have	Feature 3	13		
Must-Have	Feature 4	1		
Must-Have	Feature 5	5		
Must-Have	Feature 6	2		
Must-Have	Feature 7	8		
Must-Have	Feature 8	13		
Must-Have	Feature 9	5		
Must-Have	Feature 10	8		
Must-Have	Feature 11	5		
Must-Have	Feature 12	2		
Must-Have	Feature 13	2		
Must-Have	Feature 14	1		
Must-Have	Feature 15	5		
Must-Have	Feature 16	13		
Must-Have	Feature 17	8		
Should-Have	Feature 18	8		
Should-Have	Feature 19	5		
Should-Have	Feature 20	13		
Should-Have	Feature 21	2		
Should-Have	Feature 22	5		
Should-Have	Feature 23	5		
Should-Have	Feature 24	8		
Should-Have	Feature 25	13		
Should-Have	Feature 26	3		
Should-Have	Feature 27	1		
Should-Have	Feature 28	5		
Should-Have	Feature 29	3		
Should-Have	Feature 30	5		
Should-Have	Feature 31	8		
Should-Have	Feature 32	5		
Should-Have	Feature 33	2		
Should-Have	Feature 34	2		
Should-Have	Feature 35	1		
Should-Have	Feature 36	5		
Could-Have	Feature 37	13		
Could-Have	Feature 38	8		
Could-Have	Feature 39	8		
Could-Have	Feature 40	5		
Could-Have	Feature 41	13		
Could-Have	Feature 42	2		
Could-Have	Feature 43	5		
Could-Have	Feature 44	5		
Could-Have	Feature 45	8		
Could-Have	Feature 46	13		
Could-Have	Feature 47	3		
Would-Have	Feature 48	1		
Would-Have	Feature 49	5		
Would-Have	Feature 50	8		
Total Work in Story Points		300		

Release planning:

Step 1	Find No.of. Iterations/Sprints	Values	Units
	Total Work in Story Points	300	Story Points
	Team Velocity	20	Story Points/Iteration
	No.of. Iterations(Sprints) =	15	Count
	Total Work in Story Points / Team Velocity		

Step 2	Project Duration Calculation		
	Iteration/Sprint Duration	2	Weeks
	Project Duration =	30	Weeks
	No.of. Iterations X Iteration Duartion		

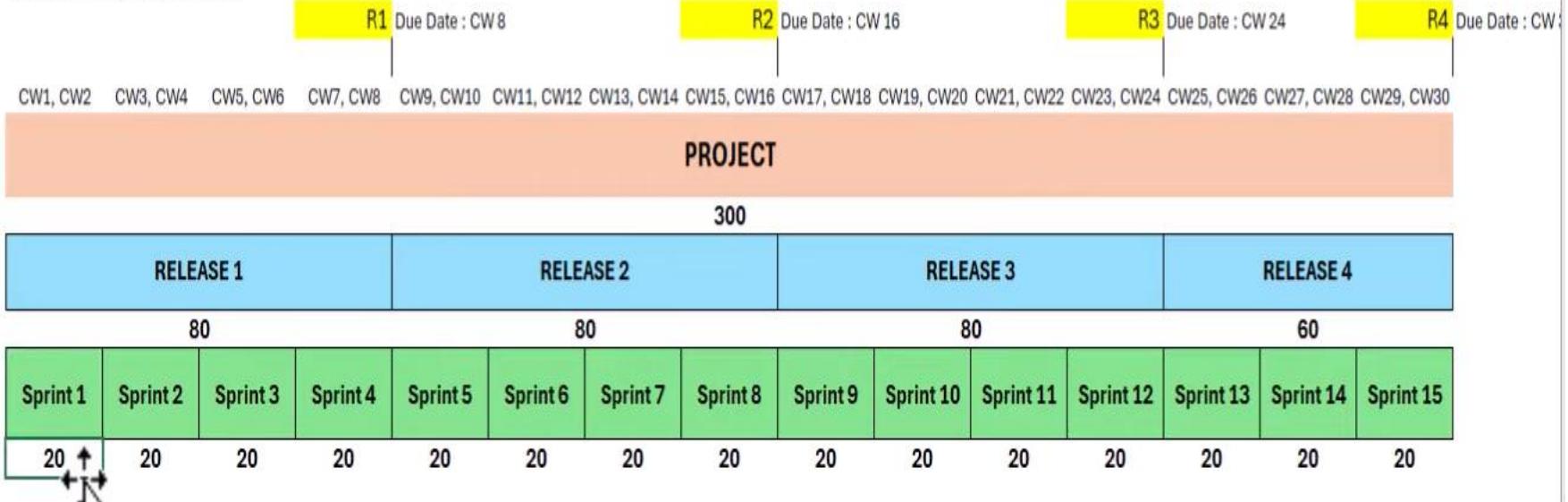
Step 3	No.of.Iterations Per Release		
	Duration of a Release	2	Months
	Duration of a Release in Weeks	8	Weeks
	No.of.Iterations Per Release	4	Count

Step 4	No.of. Releases Need in the Project		
	No.of. Iterations (Sprints)	15	Count
	No.of.Iterations Per Release	4	Count
	No.of. Releases Need in the Project	3.75	Count

Priority	Scope	Size (Story Points)	
Must-Have	Feature 1	8	
Must-Have	Feature 2	5	
Must-Have	Feature 3	13	
Must-Have	Feature 4	1	
Must-Have	Feature 5	5	
Must-Have	Feature 6	2	
Must-Have	Feature 7	8	
Must-Have	Feature 8	13	
Must-Have	Feature 9	5	
Must-Have	Feature 10	8	
Must-Have	Feature 11	5	
Must-Have	Feature 12	2	
Must-Have	Feature 13	2	
Must-Have	Feature 14	1	
Must-Have	Feature 15	5	
Must-Have	Feature 16	13	
Must-Have	Feature 17	8	
Should-Have	Feature 18	8	
Should-Have	Feature 19	5	
Should-Have	Feature 20	13	
Should-Have	Feature 21	2	
Should-Have	Feature 22	5	
Should-Have	Feature 23	5	
Should-Have	Feature 24	8	
Should-Have	Feature 25	13	
Should-Have	Feature 26	3	
Should-Have	Feature 27	1	
Should-Have	Feature 28	5	
Should-Have	Feature 29	3	
Should-Have	Feature 30	5	
Should-Have	Feature 31	8	
Should-Have	Feature 32	5	
Should-Have	Feature 33	2	
Should-Have	Feature 34	2	
Should-Have	Feature 35	1	
Should-Have	Feature 36	5	
Could-Have	Feature 37	13	
Could-Have	Feature 38	8	
Could-Have	Feature 39	8	
Could-Have	Feature 40	5	
Could-Have	Feature 41	13	
Could-Have	Feature 42	2	
Could-Have	Feature 43	5	
Could-Have	Feature 44	5	
Could-Have	Feature 45	8	
Could-Have	Feature 46	13	
Could-Have	Feature 47	3	
Would-Have	Feature 48	1	
Would-Have	Feature 49	5	
Would-Have	Feature 50	8	
Total Work in Story Points		300	

Release planning:

RELEASE PLAN - FIXED DATE



What is Velocity in Agile?

- **Velocity** is an empirical measure of how much work a team completes in one sprint, usually expressed as the sum of story points for all user stories (or backlog items) that meet the team's Definition of Done by the end of the sprint.
- It's used for **short-term forecasting** (how many sprints to finish a backlog) and to help set a realistic scope for future sprints.

1. Understand what velocity means in Scrum

- Velocity = Total story points of completed backlog items in a sprint.
- Only count items that meet the Definition of Done (DoD).
- It's measured per team, not per person

2. Steps to calculate velocity

- **Step 1 – Choose a measurement Unit**
- Usually story points (relative effort estimates).
- Could be ideal days or other agreed-upon units, but story points are common.

Step 2 – At sprint end, sum up the completed work

- Look at all user stories completed in the sprint.
- Add their story points together.
- Example: Sprint 1 completed stories:
- Login (8 points)
- View reports (5 points)
- Export to Excel (7 points)
- Total velocity for Sprint 1 = **$8 + 5 + 7 = 20$ story points.**

- **Step 3 – Track for several sprints**
- Velocity can vary from sprint to sprint because of:
 - Different scope
 - Team availability
 - Complexity of work
- Record it for at least **3–5 sprints** to get a stable average.

- Step 4 – Calculate average velocity
- Example over 4 sprints:
- Sprint 1 → 20 points
- Sprint 2 → 18 points
- Sprint 3 → 22 points
- Sprint 4 → 20 points
-
- $\text{Total} = 20 + 18 + 22 + 20 = \mathbf{80 \text{ points}}$
- $\text{Average velocity} = 80 \div 4 = \mathbf{20 \text{ points per sprint.}}$

- **Step 5 – Use it for forecasting**
- If backlog total = **100 story points**
- Average velocity = **20 points/sprint**
- Estimated sprints needed = $100 \div 20 = 5$ sprints.

3. Formula

Velocity = $\frac{\text{Sum of story points completed in the sprint}}{\text{Number of sprints (for average)}}$

Note:

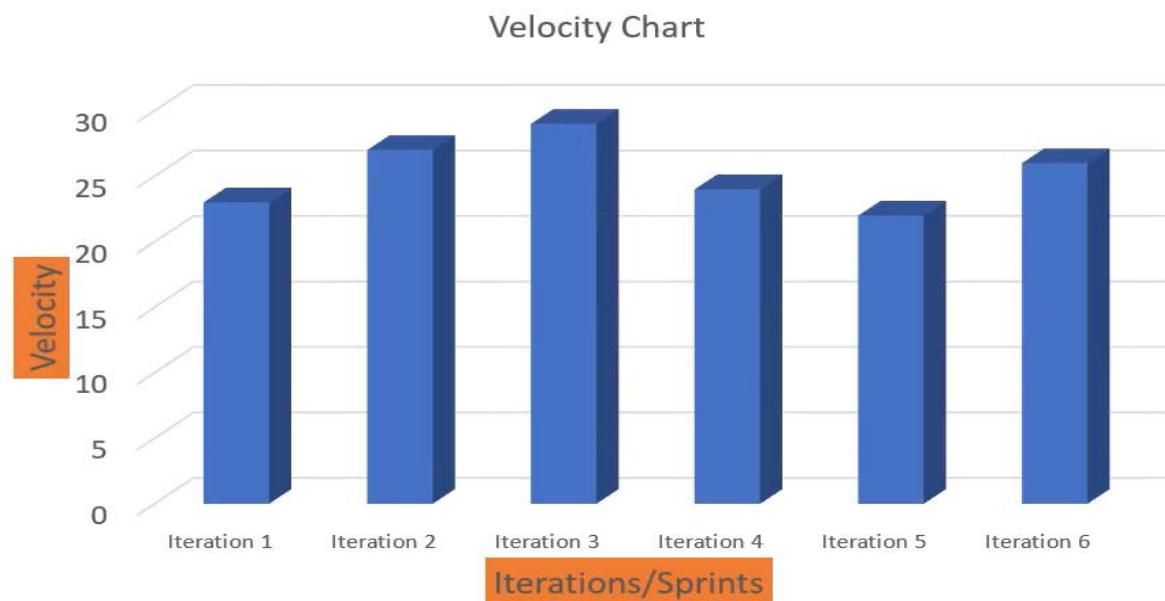
- Count **only completed items**.
- Don't compare velocity **between teams** (each team estimates differently).
- Don't push teams to “increase velocity” artificially — it's a planning tool, not a performance score.
- Use **rolling averages** for better predictions.

Velocity

Finding Average Velocity

To do an Iterative Schedule we should know the Team's Average Velocity (Actual/ Stable velocity)

To find the Team's Average Velocity – We should conduct 6 Iterations/Sprints



Iterations	Velocity
Iteration 1	23
Iteration 2	27
Iteration 3	29
Iteration 4	24
Iteration 5	22
Iteration 6	26
Average	25



What is RoadMaps?

- A **visual timeline** of major product releases, features, and goals.
- Aligns stakeholders, development teams, and customers around the same vision.
- Flexible — can be adjusted as priorities and market conditions change.
- Helps in **long-term planning** while still supporting Agile's adaptability.

ROADMAP

	Q1	Q2	Q3
Authentication	Release 1		
User Profile	Release 1		
Notification Preferences		Release 2	
Reporting	Release 2		Release 3

Types of Agile Roadmaps

Type	Description	When to Use
Strategic Roadmap	High-level goals aligned with business objectives.	Early planning to set product direction.
Release Roadmap	Shows planned releases and timelines for each.	To coordinate releases across teams.
Feature Roadmap	Focused on specific features and their delivery schedule.	When stakeholders care about feature delivery.
Technology Roadmap	Focuses on infrastructure, tech stack, upgrades, and technical debt.	For engineering & architecture planning.
Market/Portfolio Roadmap	Shows multiple products or initiatives in one view.	For companies with multiple product lines.
Theme-based Roadmap	Groups work into high-level themes instead of features/dates.	When goals matter more than exact deadlines.

What is Capacity Planning?

- Capacity Planning is the process of determining the amount of work a team can complete during a given time frame (usually a sprint), considering the team's availability, skills, and other constraints.
- It helps ensure realistic sprint commitments and balanced workloads.

Agile Planning and Estimation



	A	B	C	D	E	F	G	H
1	Sprint No	2						
2	Sprint Start Date	01-01-2021	Sprint End Date	14-01-2021	Capacity	Previous Sprint	Current Sprint	
3	Duration(Exclude Sat and Sun)	10			Estimated	30	33	
4	Public Holidays	1	Hours per story point	10	Achieved	30		
5	Total available working Days	9	Working Hours per day	8				
6	Max Hours available per member	72						
7								
8								
9	Team Members	Designation	Allocation for Team (%)	Scrum Ceremonies (%)	Learning and Development (%)	Planned Leaves(Hours)	Actual Availability	Consider
10	Peter	Developer	100	10	20	8	42	Yes
11	John	Developer	100	10	10	0	58	Yes
12	Donald	Developer	100	10	10	0	58	Yes
13	Mary	Developer	100	10	10	0	58	Yes
14	Christe	Developer	100	10	10	0	58	Yes
15	sonia	QA	100	10	10	0	58	Yes
16						Total availability in Hours	332	
17								
18	Joseph	Product Owner	100	10	10	0	58	No
19	Philips	Business Analyst	100	10	10	0	58	No
20								

ACTUAL AVAILABILITY OF EACH MEMBER :

Actual Availability of Peter assuming that he is allocated 100 percent to Team

Sprint Duration	14 Days
Weekend Holidays	$14 - 4 = 10$ days
Public Holidays	1 day (8 hours)
Max available Hours in this sprint for each member	$9 * 8 = 72$ Hours

Deduction Factors	Deduction in Hours
Time spent for Learning and Development – 15 %	15 % of 72 Hours = 10.8 Hours
Time spent for attending Scrum Ceremonies – 10%	10 % of 72 Hours = 7.2 Hours
Planned Leaves – 1 day	8 Hours
Total Deduction	$(10.8 + 7.2 + 8) = 26$ Hours

Actual Hours available for Peter in sprint 2
Maximum available Hours – Total Deduction of Peter
72 Hours - 26 Hours = 46 Hours

Agile Planning and Estimation

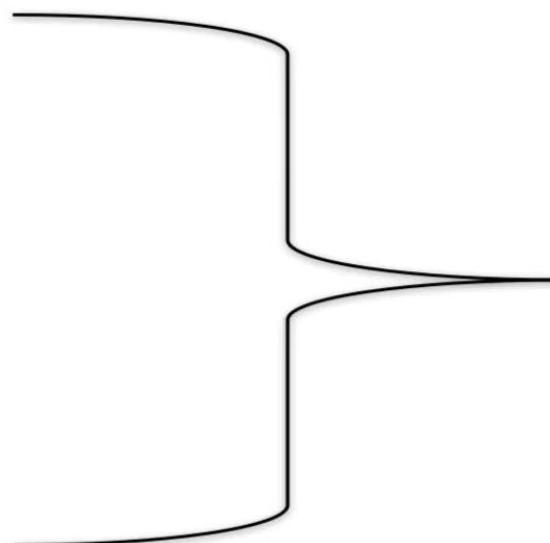


A	B	C	D	E	F	G	H
1 Sprint No	2						
2 Sprint Start Date	01-01-2021	Sprint End Date	14-01-2021	Capacity	Previous Sprint	Current Sprint	
3 Duration(Exclude Sat and Sun)	10			Estimated	30	32	
4 Public Holidays	1	Hours per story point	10	Achieved	30		
5 Total available working Days	9	Working Hours per day	8				
6 Max Hours available per member	72						
7							
8							
9 Team Members	Designation	Allocation for Team (%)	Scrum Ceremonies (%)	Learning and Development (%)	Planned Leaves(Hours)	Actual Availability	Consider
10 Peter	Developer	100	10	20	8	42	Yes
11 John	Developer	100	10	10	0	58	Yes
12 Donald	Developer	100	10	10	0	58	Yes
13 Mary	Developer	100	10	10	0	58	Yes
14 Christe	Developer	100	10	10	0	58	Yes
15 sonia	QA	80	10	10	0	46	Yes
16					Total availability in Hours	320	
17							
18 Joseph	Product Owner	100	10	10	0	58	No
19 Philips	Business Analyst	100	10	10	0	58	No

CAPACITY PLANNING

Product Backlog

User Stories	Estimation
User Story 1	8 points
User Story 2	8 points
User Story 3	4 points
User Story 4	8 points
User Story 5	4 points
User Story 6	13 points
User Story 7	8 points
User Story 8	8 points
User Story 9	4 points



Team Capacity – 32 story points

Sprint Backlog

User stories	Story Points
User Story 1	8 points
User Story 2	8 points
User Story 3	4 points
User Story 4	8 points
User Story 5	4 points
Total Points	32 points

